## **CSE 417: Algorithms and Computational Complexity**

6: Dynamic Programming, III Longest Increasing Subseq.

> Winter 2005 W. L. Ruzzo

> > 1

3

## Three Steps to Dynamic Programming

- Formulate the answer as a recurrence relation or recursive algorithm
- Show that number of different parameters in the recursive algorithm is "small" (e.g., bounded by a low-degree polynomial)
- Specify an order of evaluation for the recurrence so that already have the partial results ready when you need them.

2

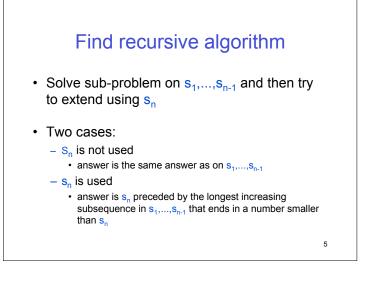
4

## Longest Increasing Run

- Given a sequence of integers S<sub>1</sub>,...,S<sub>n</sub> find a subsequence S<sub>i</sub> < S<sub>i+1</sub> <... < S<sub>i+k</sub> so that k > 0 is as large as possible.
- e.g. Given 9,5,2,5,8,7,3,1,6,9 as input,
  - possible increasing subsequence is 1,6
  - better is 2,5,8 or 1,6,9 (either or which would be a correct output to our problem)

Longest Increasing Subsequence

- Given a sequence of integers S<sub>1</sub>,...,S<sub>n</sub> find a subsequence S<sub>i1</sub> < S<sub>i2</sub> <... < S<sub>ik</sub> with i1 <... <i k so that k is as large as possible.</li>
- e.g. Given 9,5,2,5,8,7,3,1,6,9 as input,
  - possible increasing subsequence is 2,5,7
  - better is 2,5,6,9 or 2,5,8,9 (either or which would be a correct output to our problem; and there are others)

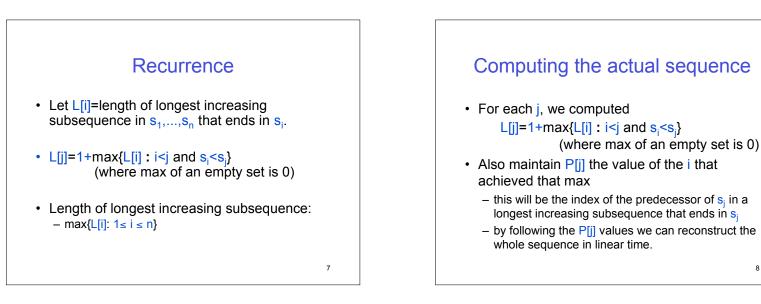


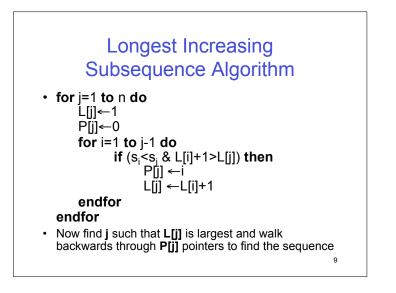
## Refined recursive idea (stronger notion of subproblem)

- Suppose that we knew for each i<n the</li> longest increasing subsequence in s1,...,sn-1 ending in S<sub>i</sub>.
  - i=n-1 is just the n-1 size sub-problem we tried before.
- Now to compute value for i=n find
  - $-s_n$  preceded by the maximum over all i<n such that si<sn of the longest increasing subsequence ending in si
- · First find the best length rather than trying to actually compute the sequence itself.

6

8





Example									
i	1	2	3	4	5	6	7	8	9
s <sub>i</sub>	90	50	20	80	70	30	10	60	40
I,									
p <sub>i</sub>									
									10